# A data-driven test of a quantum-statistics PDF parametrisation

Marco Bonvini, Federico Silvetti

INFN, Rome 1 unit    Sapienza University of Rome

xFitter external meeting, CERN 3 May 2023

based on work in collaboration with F.Buccella, F.Giuli and F.Tramontano

Istituto Nazionale di Fisica Nucleare
Sezione di ROMA

# PDF determination and parametrisation choice

To fit PDFs $\rightarrow$ specify initial (low) scale $Q_0$ and PDF parametrisation

$$Q_0 \rightarrow f_i\big(x, Q_0^2, \{P\}\big)$$

No analytical way to pick the initial parametrisation nor $Q_0$

- HERAPDF $\rightarrow f_i\big(x, Q_0^2\big) = x^{\alpha}(1-x)^{\beta} P_n(z)$ using polynomials [1506.06042]
- NNPDF $\rightarrow f_i\big(x, Q_0^2\big) = x^{\alpha}(1-x)^{\beta} \mathrm{NN}(z)$ using neural networks [2109.02653]
- BG $\rightarrow f_i\big(x, Q_0^2\big) = x^{\alpha}(1-x)^{\beta} [P_n(z) + P_m(\log(z))]$ using $\log(z)$ polynomial for small-$z$ region [1902.11125]
- $\cdots$ many others [2207.04739],[1912.10053]

An alternative to a generic parametrisation is using physical arguments to model the structures of the proton...

# Quantum Statistical Model

**Proton** $\rightarrow$ gas mixture of massless partons **at equilibrium** [1412.7683]

$$f^{\uparrow\downarrow}(E) = \frac{g_f V}{(2\pi)^3}\left[\exp\left(\frac{(E-\mu_f^{\uparrow\downarrow})}{T}\right) \pm 1\right]^{-1},$$

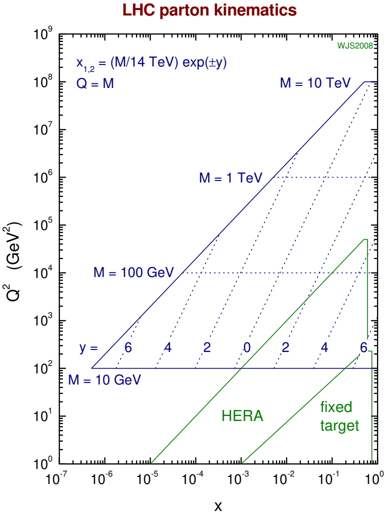
In principle, constrain $\{V, T, \mu_f^{\uparrow\downarrow}\}$ using sum rules+maximising entropy
For a recent example Phys. Lett. B 775 (2017) 172 – 177
Alternatevely, modify the distribution to build a PDF parametrisation

**Proton** → gas mixture of massless partons **at equilibrium** [1412.7683]

- rewrite using dimensionless parameters

$$f_i^{\uparrow\downarrow}(x, Q_0^2) \supset \left[\exp\left(\frac{x - X_{0i}^{\uparrow\downarrow}}{\bar{x}}\right) \pm 1\right]^{-1};$$

- "Chemical potentials" of quarks and antiquarks are related $X_{0q}^{\uparrow\downarrow} = -X_{0\bar{q}}^{\downarrow\uparrow}$
- Gluons behave like blackbody radiation → $X_{0g} = 0$
- Replace normalisation factor with
  $Ax^b X_{0i}^{\uparrow\downarrow}$ for quark
  $\bar{A}x^{\bar{b}}/X_{0i}^{\downarrow\uparrow}$ for antiquark
- sum rules should imply $u^\uparrow > d^\downarrow > u^\downarrow > d^\uparrow$ which would lead to $X_{0u}^\uparrow > X_{0d}^\downarrow > X_{0u}^\downarrow > X_{0d}^\uparrow$



# QS PDF determination

Various determinations of QS PDF parameters in the literature

So far, determinations are mostly based on fits against public PDFs

[hep-ph/0109160] [1412.7683] [1502.02517] [2201.07640]

We want to perform a legitimate PDF fits to data using this QS parametrization

- to test the model
- (more important) to explore a new PDF parametrization depending on few parameters (useful e.g. for assessing parametrization bias)

# Our QSPDF parametrisation (1)

We summarize the expressions from the model as [1412.7683]

$$h(x;b,\bar{x},X) = \frac{x^b}{\exp\left(\frac{x-X}{\bar{x}}\right)+1},$$

$$xf_q^{\uparrow\downarrow}(x,Q_0^2) = AX_q^{\uparrow\downarrow}h\left(x;b,\bar{x},X_q^{\uparrow\downarrow}\right) + \tilde{A}h\left(x;\tilde{b},\bar{x},0\right), \tag{1a}$$

$$xf_{\bar{q}}^{\uparrow\downarrow}(x,Q_0^2) = \bar{A}\frac{1}{X_q^{\downarrow\uparrow}}h\left(x;\bar{b},\bar{x},-X_q^{\downarrow\uparrow}\right) + \tilde{A}h\left(x;\tilde{b},\bar{x},0\right), \tag{1b}$$

$$\text{with} \quad q \in \{u,\, d\},$$

$$xf_g(x,Q_0^2) = \frac{A_g x^{b_g}}{\exp(x/\bar{x})-1}. \tag{1c}$$

An auxiliary "diffractive" term $\tilde{A}h\left(x;\tilde{b},\bar{x},0\right)$ is introduced to control the high-energy region.

Fitting only unpolarised DIS data → sum over helicity

$$f_q(x,Q_0^2) = f_q^{\uparrow}(x,Q_0^2) + f_q^{\downarrow}(x,Q_0^2)$$

Writing the unpolarised valence and sea contributions ($q \in \{u, d\}$)

$$\begin{aligned} xf_{q_v}(x, Q_0^2) &= q(x, Q_0^2) - \bar{q}(x, Q_0^2) \\ &= A\left[X_q^{\uparrow} h\left(x; b, \bar{x}, X_q^{\uparrow}\right) + X_q^{\downarrow} h\left(x; b, \bar{x}, X_q^{\downarrow}\right)\right] \\ &\quad - \bar{A}\left[\frac{1}{X_q^{\downarrow}} h\left(x; \bar{b}, \bar{x}, -X_q^{\downarrow}\right) + \frac{1}{X_q^{\uparrow}} h\left(x; \bar{b}, \bar{x}, -X_q^{\uparrow}\right)\right] \end{aligned} \tag{2a}$$

$$\begin{aligned} xf_{\bar{q}}(x, Q_0^2) &= \bar{A}\left[\frac{1}{X_q^{\downarrow}} h\left(x; \bar{b}, \bar{x}, -X_q^{\downarrow}\right) + \frac{1}{X_q^{\uparrow}} h\left(x; \bar{b}, \bar{x}, -X_q^{\uparrow}\right)\right] \\ &\quad + 2\tilde{A} h\left(x; \tilde{b}, \bar{x}, 0\right), \end{aligned} \tag{2b}$$

$$xf_g(x, Q_0^2) = \frac{A_g x^{b_g}}{\exp(x/\bar{x}) - 1} \tag{2c}$$

$$s(x, Q_0^2) = \bar{s}(x, Q_0^2) = \frac{f_s}{1 - f_s} \bar{d}(x, Q_0^2) \quad \text{with} \quad f_s = 0.4\,. \tag{2d}$$

There are 13 parametres: $\left\{\bar{x}, A_g, A, \bar{A}, \tilde{A}, X_u^{\uparrow\downarrow}, X_d^{\uparrow\downarrow}, b, \bar{b}, b_g, \tilde{b}\right\}$

# Constraints and other setups

Additional constraints we apply:

- Valence and momentum **sum rules** $\to$ fix normalisations $\{A_g, A, \bar{A}\}$
  Valence sum rules require some care ($A, \bar{A}$ are not overall factors) $\to$ see later
- Regge theory + DGLAP imply that gluon and quark singlet behave the same at small $x$

$$xf_q(x, Q_0^2) \sim x^{\tilde{b}} \tag{3}$$

$$xf_g(x, Q_0^2) \sim x^{b_g - 1} \tag{4}$$

  which implies the constraint

$$b_g = 1 + \tilde{b}$$

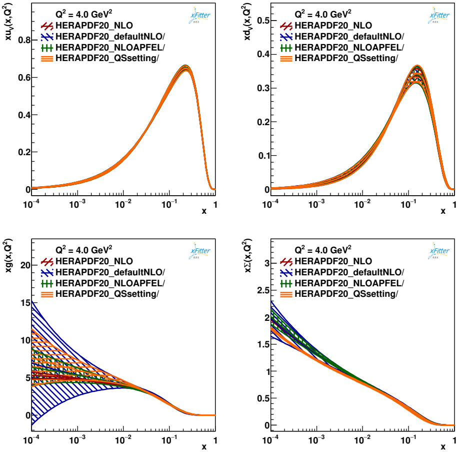
- We also fix $\bar{b} = b$ for simplicity (further studies are ongoing)

This leaves $8$ free parameters to fit: $\left\{\bar{x}, \tilde{A}, X_u^{\uparrow\downarrow}, X_d^{\uparrow\downarrow}, b, \tilde{b}\right\}$.

Potential bug found in NLO fit (gitlab master branch **84c362f1** 17/11/2022)

**Legend:**

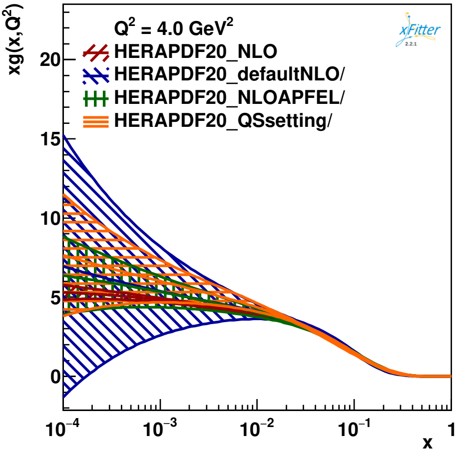
- HERAPDF2.0_NLO plot of the publicly available LHAPDF set
- HERAPDF2.0_defaultNLO output of the example in `xfitter`
- HERAPDF2.0_NLOAPFEL same as above, using different theory input
- HERAPDF2.0_QSetting benchmark for our parametrisation (wait next slides)



Potential bug found in NLO fit (gitlab master branch **84c362f1** 17/11/2022)

**Legend:**

- HERAPDF2.0_NLO plot of the publicly available LHAPDF set
- HERAPDF2.0_defaultNLO output of the example in `xfitter`
- HERAPDF2.0_NLOAPFEL same as above, using different theory input
- HERAPDF2.0_QSetting benchmark for our parametrisation (wait next slides)

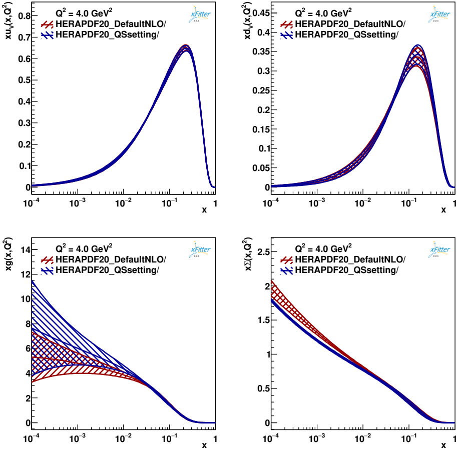


# Fit setup and benchmark

Benchmark fit between default HERAPDF2.0 NLO configuration in `xfitter` and our settings

Some additional constraints are applied:

- Parametrisation scale $Q_0^2 = 4$ GeV$^2$
  $\rightarrow$ we have to cut out the $Q^2 = 3.5$ GeV$^2$ bin
- Theory inputs:
  `APFEL@NLO`
  VFNS (FONLL-B)



# Fit setup and benchmark

Benchmark fit between default HERAPDF2.0 NLO configuration in xfitter and our settings

Some additional constraints are applied:

- Parametrisation scale $Q_0^2 = 4$ GeV$^2$
  $\rightarrow$ we have to cut out the $Q^2 = 3.5$ GeV$^2$ bin
- Theory inputs:
  APFEL@NLO
  VFNS (FONLL-B)
- Improved description of NCep 920

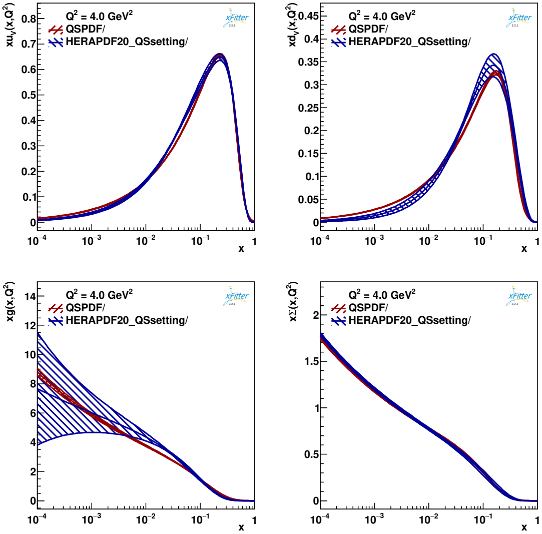
| Dataset | HERAPDF20 Default-NLO | HERAPDF20 QSsetting |
|---|---|---|
| HERA1+2 CCem | 54 / 42 | 54 / 42 |
| HERA1+2 NCep 820 | 68 / 70 | 64 / 68 |
| HERA1+2 NCep 460 | 217 / 204 | 216 / 200 |
| → HERA1+2 NCep 920 | 439 / 377 | 397 / 363 ← |
| HERA1+2 CCep | 43 / 39 | 45 / 39 |
| HERA1+2 NCem | 222 / 159 | 221 / 159 |
| HERA1+2 NCep 575 | 219 / 254 | 217 / 249 |
| Correlated $\chi^2$ | 86 | 67 |
| Log penalty $\chi^2$ | +8.3 | -4.68 |
| Total $\chi^2$ / dof | 1357 / 1131 | 1275 / 1106 |
| $\chi^2$ p-value | 0.00 | 0.00 |

# Testing the QSPDF parametrisation: fits and $\chi^2$

Now we fit QSPDF against the HERA DIS dataset.

- Reduced error bands
- Only experimental uncertainty is accounted for

In blue the benchmark fit with the HERAPDF2.0 and in red the fit of QSPDF



# Testing the QSPDF parametrisation: fits and $\chi^2$

Now we fit QSPDF against the HERA DIS dataset.

- Fit quality is good
- very minimal set of parametres

| Dataset | QSPDF | HERAPDF20 QSsetting |
|---|---|---|
| HERA1+2 CCep | 59 / 39 | 45 / 39 |
| HERA1+2 CCem | 69 / 42 | 54 / 42 |
| HERA1+2 NCem | 229 / 159 | 221 / 159 |
| HERA1+2 NCep 820 | 71 / 68 | 64 / 68 |
| HERA1+2 NCep 920 | 468 / 363 | 397 / 363 |
| HERA1+2 NCep 460 | 231 / 200 | 216 / 200 |
| HERA1+2 NCep 575 | 235 / 249 | 217 / 249 |
| Correlated $\chi^2$ | 104 | 67 |
| Log penalty $\chi^2$ | -71.03 | -4.68 |
| Total $\chi^2$ / dof | 1397 / 1112 | 1275 / 1106 |
| $\chi^2$ p-value | 0.00 | 0.00 |

| | QSPDF | HERAPDF2.0 |
|---|---|---|
| # param. | 8 | 14 |
| $\chi^2$/D.O.F. | 1.26 | 1.15 |

Comparison with older determinations (not real fits to data) [hep-ph/0109160]

We find:

- $X_u^\uparrow > X_d^\downarrow \sim X_u^\downarrow > X_d^\uparrow$
- Qualitative agreement determination of parameters
- No information on polarised PDF (unlike previous work)

| Parameter | QSPDF | [hep-ph/0109160] |
|---|---|---|
| $A$ | $3.04$ | 1.75 |
| $\bar{A}$ | $0.12$ | 1.91 |
| $A_g$ | $33.52$ | 14.28 |
| $\tilde{A}$ | $0.133 \pm 0.004$ | 0.083 |
| $X_d^\uparrow$ | $0.14 \pm 0.02$ | 0.23 |
| $X_d^\downarrow$ | $0.284 \pm 0.007$ | 0.302 |
| $X_u^\uparrow$ | $0.419 \pm 0.007$ | 0.461 |
| $X_u^\downarrow$ | $0.21 \pm 0.02$ | 0.298 |
| $b = \bar{b}$ | $0.52 \pm 0.01$ | 0.41 |
| $\tilde{b} = b_g - 1$ | $-0.173 \pm 0.003$ | -0.253 |
| $\bar{x}$ | $0.092 \pm 0.001$ | 0.099 |

# Testing the QSPDF parametrisation: $\bar{d} - \bar{u}$ distribution

We compare the $\bar{d} - \bar{u}$ distributions explicitly
$\rightarrow$ Interesting qualitative feature reproduced!

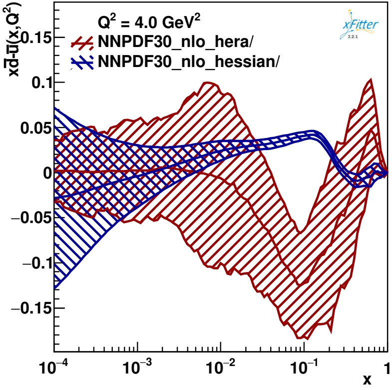



Figure: Comparison of NNNPDF3.0 fits with the HERA dataset only and the default dataset, (NLO theory)

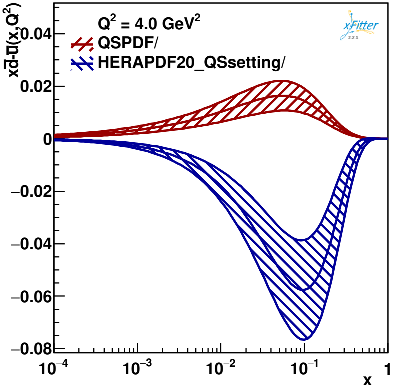



Figure: In blue the benchmark fit with the HERAPDF2.0 parametrisation and in red the fit of QSPDF

# Conclusions and outlook

## Fitting a statistical PDF model

- ✓ We performed a fit a custom PDF parametrisation (`QSPDF` ) against the HERA DIS dataset
- ↑ Acceptable agreement between data and model
- ↑ Fit parameters match a previous attempts at fitting a similar parametrisation
- ↓ This simplest iteration of the parametrisation isn't very competitive against more established models...
- ↑ ...but uses a smaller number of degrees of freedom (8 pars vs 14 of HERAPDF)
- ↑ Qualitative shape of the $\bar{d} - \bar{u}$ distribution reproduced with HERA data only

# Conclusions and outlook

## Fitting a statistical PDF model

- ✓ We performed a fit a custom PDF parametrisation (`QSPDF` ) against the HERA DIS dataset
- ↑ Acceptable agreement between data and model
- ↑ Fit parameters match a previous attempts at fitting a similar parametrisation
- ↓ This simplest iteration of the parametrisation isn't very competitive against more established models...
- ↑ ...but uses a smaller number of degrees of freedom (8 pars vs 14 of HERAPDF)
- ↑ Qualitative shape of the $\bar{d} - \bar{u}$ distribution reproduced with HERA data only

## Outlook

- Improve the current fit (tuning $Q_0$, NNLO theory, resummed theory)
- Modify the parametrisation (relaxing $\bar{b}$, add transverse potentials) $\rightarrow$ building a "minimal" parameter set with state-of-the art performance
- Possibility to fit simultaneously unpolarized and polarized PDFs in terms of the same parameters

# Thank you for your attention...

# but it's not over!

Now it's time for technicalities...

# Sum rules

Quark number and momentum sum rules are used to fix three parameters

$$2 = \int_0^1 dx\, u_v(x,\mu^2), \quad 1 = \int_0^1 dx\, d_v(x,\mu^2), \quad 1 = \int_0^1 dx\, x \left[ g(x,\mu^2) + \sum_q f_q(x,\mu^2) \right]$$

Usually, these are the normalizations of $u_v$, $d_v$ and $g$.

OK for the gluon: momentum sum rule fixes $A_g$ in $xg(x,Q_0^2) = A_g x^{b_g}/(\exp(x/\bar{x}) - 1)$.

However, here $u_v$ and $d_v$ do not have overall normalizations!

$$\begin{aligned} xu_v(x,Q_0^2) &= A\Big[X_u^\uparrow h\Big(x;b,\bar{x},X_u^\uparrow\Big) + X_u^\downarrow h\Big(x;b,\bar{x},X_u^\downarrow\Big)\Big] \\ &\quad - \bar{A}\left[\frac{1}{X_u^\downarrow} h\Big(x;\bar{b},\bar{x},-X_u^\downarrow\Big) + \frac{1}{X_u^\uparrow} h\Big(x;\bar{b},\bar{x},-X_u^\uparrow\Big)\right] \\ xd_v(x,Q_0^2) &= A\Big[X_d^\uparrow h\Big(x;b,\bar{x},X_d^\uparrow\Big) + X_d^\downarrow h\Big(x;b,\bar{x},X_d^\downarrow\Big)\Big] \\ &\quad - \bar{A}\left[\frac{1}{X_d^\downarrow} h\Big(x;\bar{b},\bar{x},-X_d^\downarrow\Big) + \frac{1}{X_d^\uparrow} h\Big(x;\bar{b},\bar{x},-X_d^\uparrow\Big)\right] \end{aligned}$$

But $A$ and $\bar{A}$ are the *same for up and down* $\rightarrow$ we can fix them by solving an algebraic system

Quark number sum rules lead to the system

$$\begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} K_u & \bar{K}_u \\ K_d & \bar{K}_d \end{pmatrix} \begin{pmatrix} A \\ \bar{A} \end{pmatrix}$$

with

$$K_q = \int_0^1 \mathrm{d}x \left[ X_q^{\uparrow} h\left(x; b, \bar{x}, X_q^{\uparrow}\right) + X_q^{\downarrow} h\left(x; b, \bar{x}, X_q^{\downarrow}\right) \right] ,$$

$$\bar{K}_q = -\int_0^1 \mathrm{d}x \left[ \frac{1}{X_q^{\downarrow}} h\left(x; \bar{b}, \bar{x}, -X_q^{\downarrow}\right) + \frac{1}{X_q^{\uparrow}} h\left(x; \bar{b}, \bar{x}, -X_q^{\uparrow}\right) \right] ,$$

Not straightforward to implement in `xFitter`!

To do so, we implemented a new `PDF decomposition` in which $u, \bar{u}, d, \bar{d}$ are treated separately
(to be precise, we define them without the "diffractive term" which is added afterwards)

# PDF decomposition (vs UvDvUbarDbarS.cc)

```
//____________________________________________________________________
void UvDvUbarDbarS::atStart(){
  const YAML::Node node=XFITTER_PARS::getDecompositionNode(_name);
  //TODO: handle errors
  par_xuv  =getParameterisation(node["xuv"].as<string>());
  par_xdv  =getParameterisation(node["xdv"].as<string>());
  par_xubar=getParameterisation(node["xubar"].as<string>());
  par_xdbar=getParameterisation(node["xdbar"].as<string>());
  par_xs   =getParameterisation(node["xs"].as<string>());
  par_xg   =getParameterisation(node["xg"].as<string>());
}

void UvDvUbarDbarS::atIteration() {
  //Enforce sum rules
  // counting sum-rules for uv and dv
  par_xuv->setMoment(-1,2.0);
  par_xdv->setMoment(-1,1.0);
  // momentum sum-rule
  // quark part
  double xsumq=0;
  xsumq+=  par_xuv  ->moment(0);
  xsumq+=  par_xdv  ->moment(0);
  xsumq+=2*par_xubar->moment(0);
  xsumq+=2*par_xdbar->moment(0);
  xsumq+=2*par_xs   ->moment(0);
  // gluon part
  par_xg->setMoment(0,1-xsumq);
}
std::map<int,double>UvDvUbarDbarS::xfxMap(double x)const
{
  double ubar=(*par_xubar)(x);
  double dbar=(*par_xdbar)(x);
  double u=(*par_xuv)(x)+ubar;
  double d=(*par_xdv)(x)+dbar;
  double s=(*par_xs)(x);
  double g=(*par_xg)(x);
  return{
```

```
//____________________________________________________________________
void QSPDF::atStart(){
  const YAML::Node node=XFITTER_PARS::getDecompositionNode(_name);
  //TODO: handle errors
  par_xu    =getParameterisation(node["xu"].as<string>());
  par_xd    =getParameterisation(node["xd"].as<string>());
  par_xubar =getParameterisation(node["xubar"].as<string>());
  par_xdbar =getParameterisation(node["xdbar"].as<string>());
  par_xdiffr=getParameterisation(node["xdiffr"].as<string>());
  par_xs    =getParameterisation(node["xs"].as<string>());
  par_xg    =getParameterisation(node["xg"].as<string>());
}

void QSPDF::atIteration() {
  //Enforce sum rules
  // counting sum-rules for uv and dv
  double *normA   =XFITTER_PARS::gParameters.at("A");
  double *normAbar=XFITTER_PARS::gParameters.at("Abar");
  *normA    = 1.;
  *normAbar = 1.;
  double Ku = par_xu   ->moment(-1);
  double Kub=-par_xubar->moment(-1);
  double Kd = par_xd   ->moment(-1);
  double Kdb=-par_xdbar->moment(-1);
  double DetK = Ku*Kdb - Kub*Kd;
  *normA    = (2*Kdb - Kub)/DetK;
  *normAbar = (Ku - 2*Kd)/DetK;
  // momentum sum-rule
  // quark part
  double xsumq=0;
  xsumq+=  par_xu    ->moment(0);
  xsumq+=  par_xd    ->moment(0);
  xsumq+=  par_xubar->moment(0);
  xsumq+=  par_xdbar->moment(0);
  xsumq+=4*par_xdiffr->moment(0);
  xsumq+=2*par_xs    ->moment(0);
  // gluon part
  par_xg->setMoment(0,1-xsumq);
}
std::map<int,double>QSPDF::xfxMap(double x)const
{
  double diffr=(*par_xdiffr)(x);
  double ubar=(*par_xubar)(x) + diffr;
  double dbar=(*par_xdbar)(x) + diffr;
  double u=(*par_xu)(x) + diffr;
  double d=(*par_xd)(x) + diffr;
  double s=(*par_xs)(x);
  double g=(*par_xg)(x);
  return{
```

# Request to xFitter developers

Please output in the final table of parameters also the normalizations found through sume rules, ideally with uncertainties!

| Parameter | output nf4to5 | QSI |
|---|---|---|
| 'A' | 1.0000 | |
| 'Abar' | 1.0000 | |
| 'Ag' | 1.0000 | |
| 'Atil' | 0.1185 ± 0.0069 | |
| 'DbarToS' | 1.0000 | |
| 'Xdd' | 0.2741 ± 0.0098 | |
| 'Xdu' | 0.204 ± 0.026 | |
| 'Xud' | 0.4327 ± 0.0076 | |

# Numerical integral at small $x$ for the sum rules

xFitter computes numerically the integral for the sum rules from $x_0 = 10^{-6}$ to 1 (using logarithmic+linear sampling)

$$\int_0^1 dx\, x^N f(x, Q_0^2) \simeq \int_{x_0}^1 dx\, x^N f(x, Q_0^2) \qquad (N = 0, 1)$$

For the computation of the sum rules, it is important to account for the small-$x$ behaviour the parametrisation, which is generally a power behaviour

$$f(x, Q_0^2) \overset{x\to 0}{\sim} \alpha\, x^\beta$$

The integral from 0 to $x_0 = 10^{-6}$ can be better approximated by

$$\int_0^{x_0} dx\, x^N f(x, Q_0^2) \simeq \alpha\, \frac{x_0^{\beta+N+1}}{\beta + N + 1}$$

Therefore, extrapolating $\beta$ and computing $\alpha$, it is easy to complement the numerical integration above $x_0$ with the analytical approximate integration below $x_0$

Useful to avoid values of $\beta$ that make the PDF non-integrable ($\beta > -1 - N$)
For QSPDFs: $b, \bar{b} > 0$, $\tilde{b} > -1$, $b_g > 0$

# BasePdfParam.cc

```
namespace xfitter{
BasePdfParam::~BasePdfParam(){if(pars)delete[]pars;}
double BasePdfParam::moment(int iMoment)const{
        /// Numeric integration
  /// Simple rule, split log/lin spacing at xsplit=0.1

  const double xsplit = 0.1;
  const double xmin = 1e-6;
  const double xminlog = log10(xmin);
  const double xmaxlog = log10(xsplit);
  const int nlog = 100;
  const int nlin = 100;

  const double xsteplog = (xmaxlog-xminlog)/nlog;
  const double xsteplin = (1.0 - xsplit)/nlin;

  double sum = 0.;
  double x = xmin;

  // log x part:
  for (int i=0; i<nlog; i++) {
    double dx = pow(10,xminlog+(i+1)*xsteplog) - pow(10,xminlog+i*xsteplog);
    double val = (*this)(x+dx/2.)*pow(x+dx/2.,iMoment);
    x += dx;
    sum += dx*val;
  }
  // lin x part:
  for (int i=0; i<nlin; i++) {
    double dx = xsteplin;
    double val = (*this)(x+dx/2.)*pow(x+dx/2.,iMoment);
    x += dx;
    sum += dx*val;
  }

  return sum;
}
```

```
// extrapolation of the x<xmin part, assuming behaviour f~alpha*x^beta
// here we extrapolate the derivative of the log function in log scale
double logx0 = log(xmin);
double y[3];
double hh, h=log(30); // to be tuned, exp(h)>1/xmin
double f0 = (*this)(xmin);
for(int i=0; i<3; i++){
  hh = h/pow(2.,i);
  volatile double tmp = logx0+hh;  // see numerical recipies
  hh = tmp-logx0;
  y[i] = ( log( (*this)(exp(logx0 + hh) ) )-log(f0) ) /hh;
}
double beta = (y[0] - 6*y[1] + 8*y[2])/3.;
double alpha = f0/pow(xmin,beta);
// shift beta to account for Mellin factor x^N
beta += iMoment;
//if beta is too small and negative sum rules are ill-defined
//hopefully setting the output to a large number will kill the normalisation
//constants and force beta to return to nicer values
if(beta>-1) sum += alpha*pow(xmin,beta+1)/(beta+1);  // xmin*f0/(beta+1)
else { sum += std::numeric_limits<double>::max();  // to be improved...
  std::cerr<< "Beware, bad beta value"<<std::endl;
}
```

**Now it's over.**

**Thank you.**

# Backup slides

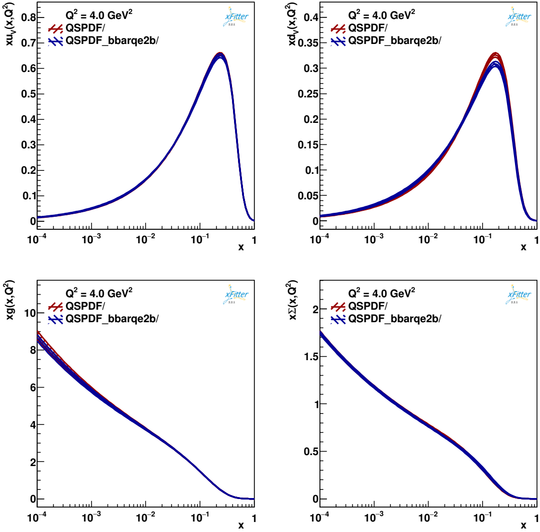



Figure: Effect of different constraint on $\bar{b}$: $\bar{b} = b$ and $\bar{b} = 2$b

# Gluon PDF error in default HERAPDF2.0 fit @NLO

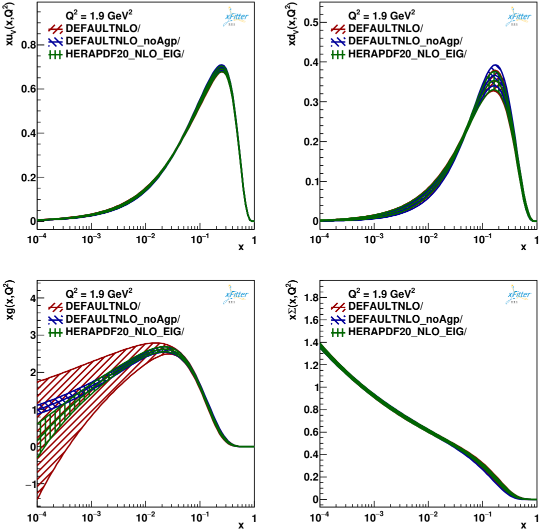



Figure: Large error in the gluon pdf induced by poor determination of $A_g' = 0.23 \pm 0.29$.